Computational Statistics
Lab 3

Group 03
Mohsen Pirmoradiyan, Ahmed Alhasan, Yash Pawar

14 Feb 2020

Question 1: Cluster sampling

An opinion pool is assumed to be performed in several locations of Sweden by sending interviewers to this location. Of course, it is unreasonable from the financial point of view to visit each city. Instead, a decision was done to use random sampling without replacement with the probabilities proportional to the number of inhabitants of the city to select 20 cities. Explore the file population.xls. Note that names in bold are counties, not cities.

1. Import necessary information to R.

```
RNGversion(min(as.character(getRversion()), "3.6.2"))
set.seed(12345, kind="Mersenne-Twister", normal.kind="Inversion")

##Question 1: Cluster sampling
population <- read.csv2("../Data/population.csv", stringsAsFactors = FALSE)
```

2. Use a uniform random number generator to create a function that selects 1 city from the whole list by the probability scheme offered above (do not use standard sampling functions present in R).

3. . Use the function you have created in step 2 as follows:

(a) Apply it to the list of all cities and select one city
(b) Remove this city from the list
(c) Apply this function again to the updated list of the cities
(d) Remove this city from the list
(e) ... and so on until you get exactly 20 cities.

```
sampler <- function(data, n){
  #taking the cumsum for the probabilities means
  #we put them in a period from 0 to 1 without overlapping
  data$prob <- cumsum(data[,2] / sum(data[,2]))

  cities <- as.character()
  pop    <- as.numeric()
  for(i in 1:n){
    rand <- runif(1)
    #applying the generalized inverse distribution function which takes only the nearest
```

```
    #CMD value that is equal or larger than the probability value ..cities with higher population
    #cover longer periods so they have higher chances to be closest to the roll
    cities[i] <- data[which.min((data$prob-rand)[which((data$prob-rand)>=0)]),][[1]]
    pop[i]    <- data[which(data[,1] == cities[i]),][[2]]
    data      <- data[-which(data[,2] == pop[i]),]
  }
  res <- data.frame(Municipality = as.character(cities), Population = as.numeric(pop))
  return(res)
}
```

4. Run the program. Which cities were selected? What can you say about the size of the selected cities?

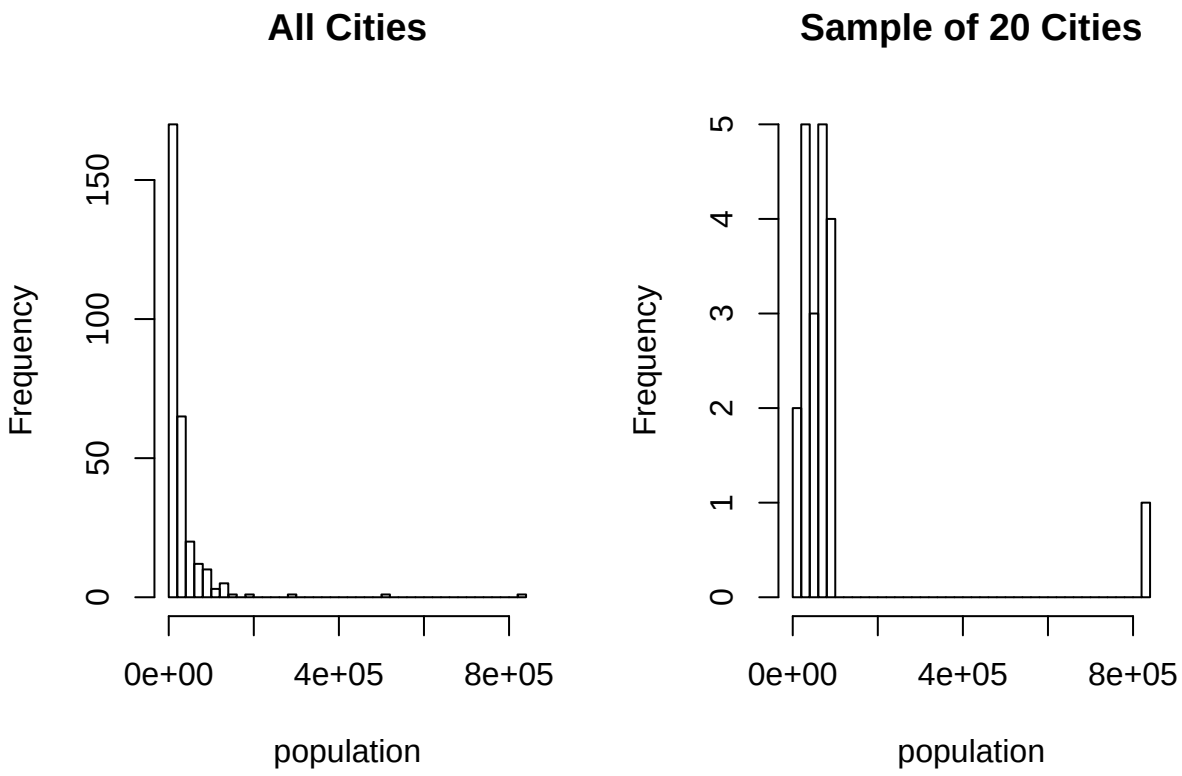
```
sampler(data = population, n = 20)
```

```
##    Municipality Population
## 1      Botkyrka      81195
## 2      Danderyd      31150
## 3         Ekerö      25095
## 4       Haninge      76237
## 5      Huddinge      95798
## 6      Järfälla      65295
## 7       Lidingö      43445
## 8         Nacka      88085
## 9     Norrtälje      55927
## 10      Nykvarn       9227
## 11    Nynäshamn      25781
## 12        Salem      15313
## 13      Sigtuna      39219
## 14   Sollentuna      63347
## 15        Solna      66909
## 16    Stockholm     829417
## 17   Sundbyberg      37722
## 18   Södertälje      85270
## 19       Tyresö      42602
## 20         Täby      63014
```

- Most of the selected cities are the ones that have larger population (higher probability)

5. Plot one histogram showing the size of all cities of the country. Plot another histogram showing the size of the 20 selected cities. Conclusions?

```
par(mfrow=c(1,2))
hist(population$Population, breaks = 50, xlab = "population", main = "All Cities")
hist(sampler(data = population, n = 20)$Population, breaks = 50, xlab = "population", main = "Sample of
```



- Since we are trying to sample from a discrete distribution we obtain the CDF by cumulatively adding up the individual probabilities for the cities and selecting the nearest city that has it's CMD value equal or larger than the probability generated from the uniform distribution of the CDF, based on the generalized inverse distribution function:

$$F^{-1}(U) = inf\{x \in \mathbb{R} : F(x) \geq U\}$$

Question 2: Different distributions

The double exponential (Laplace) distribution is given by formula:

$$DE(\mu, \alpha) = \frac{\alpha}{2} \exp(-\alpha|x - \mu|)$$

1. Write a code generating double exponential distribution DE(0,1) from Unif(0,1) by using the inverse CDF method. Explain how you obtained that code step by step. Generate 10000 random numbers from this distribution, plot the histogram and comment whether the result looks reasonable.

- From the PDF of the lablace distribution $DE(\mu, \alpha) = \frac{\alpha}{2} \exp(-\alpha|x-\mu|)$ we get the CMD by integrating.

$$\begin{aligned} F(x \geq \mu) &= \int_{-\infty}^{0} \frac{1}{2}\, exp(x)\, dx \; + \int_{0}^{x} \frac{1}{2}\, exp(-x)\, dx \\ &= \frac{1}{2}\left[exp(x)\right]_{-\infty}^{0} + \frac{1}{2}\left[-exp(-x)\right]_{0}^{x} \\ &= 1 - \frac{1}{2} exp(-x) \end{aligned}$$

$$\begin{aligned} F(x < \mu) &= \int_{-\infty}^{x} \frac{1}{2}\, exp(x)\, dx \\ &= \frac{1}{2}\left[exp(x)\right]_{-\infty}^{x} \\ &= \frac{1}{2} exp(x) \end{aligned}$$

- by substituting $F^{-1}F(x)$ in x, where we have $U = F(x)$ we get $F^{-1}(U)$

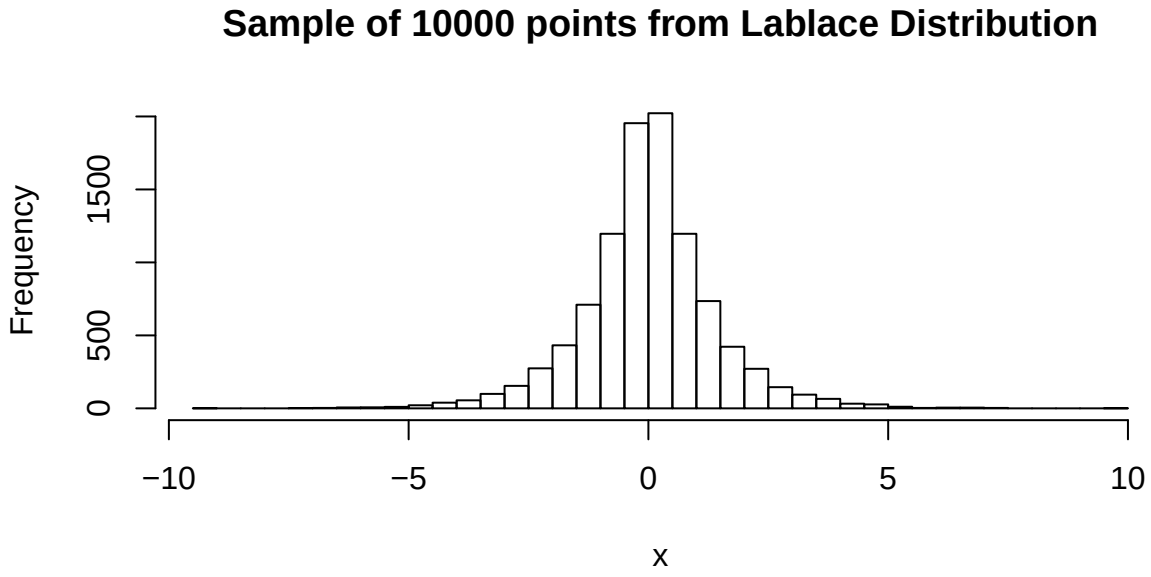
$$U = 1 - \frac{1}{2} exp(-x) \quad \Rightarrow \quad F^{-1}(U) = -ln\ 2(1-U) \quad \ldots \quad for\ U \geq \frac{1}{2}$$

$$U = \frac{1}{2} exp(x) \quad \Rightarrow \quad F^{-1}(U) = ln\ 2U \quad \ldots \quad for\ U < \frac{1}{2}$$

- Using the inverse function $F^{-1}(U)$ we can obtain the lablace probability against each uniform probability.

```
inverse_cmd <- function(n){
  u <- runif(n)

  inverse <- c()
  for(i in 1:n){
    if(u[i] < 0.5){
      inverse[i] <- log(2 * u[i])
```

```
    }
    else{
      inverse[i] <- -log(2 * (1 - u[i]))
    }
  }
  return(inverse)
}

x <- inverse_cmd(10000)
hist(x, breaks = 50, main = "Sample of 10000 points from Lablace Distribution")
```



- The resulted sample has close resemblance to the laplace distribution (double exponential) so it looks reasonable.

2. Use the Acceptance/rejection method with DE(0,1) as a majorizing density to generate N(0,1) variables. Explain step by step how this was done. How did you choose constant c in this method? Generate 2000 random numbers N(0,1) using your code and plot the histogram. Compute the average rejection rate R in the acceptance/rejection procedure. What is the expected rejection rate ER and how close is it to R? Generate 2000 numbers from N(0,1) using standard rnorm() procedure, plot the histogram and compare the obtained two histograms.

Target density function:

$$f_Y = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$$

Majorizing density function:

$$f_X = \frac{1}{2} \exp(-|x|)$$

$$cf_X \geq f_Y$$

$$c = \frac{2}{\sqrt{2\pi}} \exp(|x| - x^2/2)$$

We set c to 0 to find the value of x

$$\frac{2}{\sqrt{2\pi}}(-x + \frac{x}{|x|}) \exp(|x| - x^2/2) = 0$$

$\Rightarrow$ for $x \geq 0 \Rightarrow x = 1$ and for $x < 0 \Rightarrow x = -1$

$$c = \frac{2}{\sqrt{2\pi}} \exp(1 - 0.5) = \frac{2\sqrt{e}}{\sqrt{2\pi}}$$

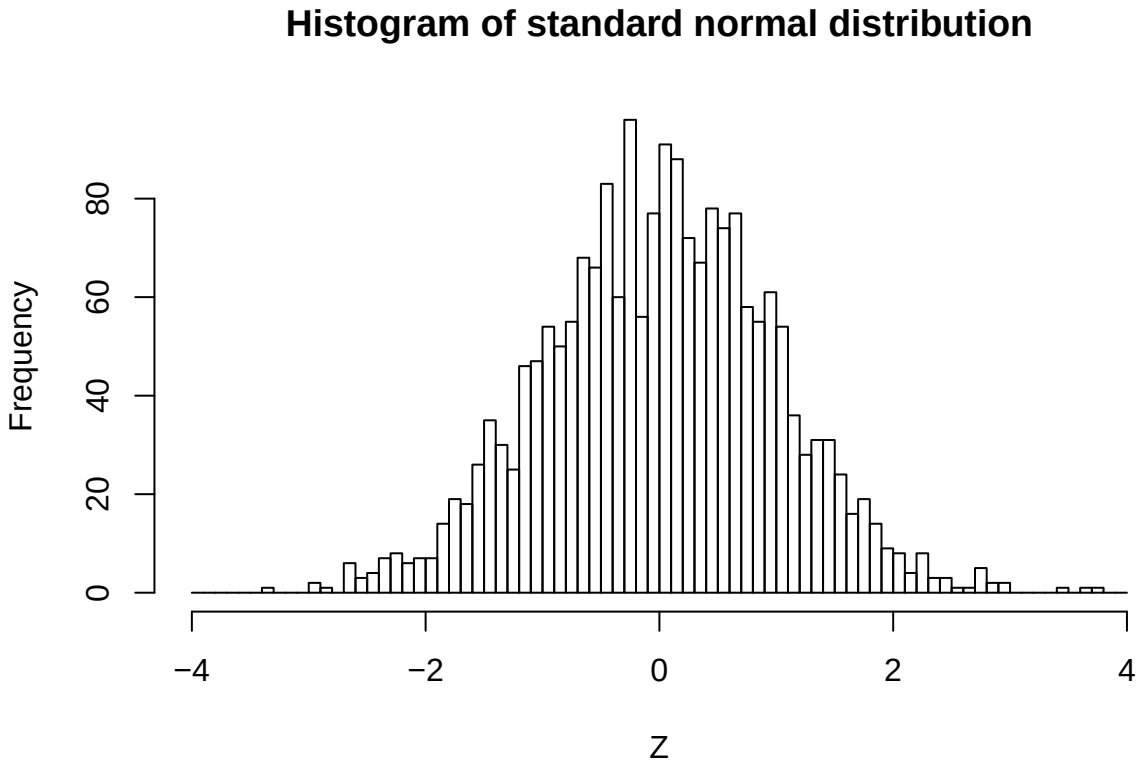
```
c = 2*sqrt(exp(1))/sqrt(2*pi)
cat("The majorizing constant:", c)
```

```
## The majorizing constant: 1.315489
```

```
# Target Density Function -- Normal Distribution
fy <- function(x){
  exp(-0.5 * x ^ 2) / sqrt(2 * pi)
}

# Majorizing Density Function -- Laplace Distribution
fx <- function(x){
  exp(-abs(x)) / 2
}

accept_reject <- function(n){
  R <- 0
  Y <- vector(length = n)
  for(i in 1:n){
    repeat {
      y <- inverse_cmd(1)
      U <- runif(1)
      h <- fy(y) / (c * fx(y))
      if(U <= h){
        Y[i] <- y
        break
        }
      else{R = R+1}
    }
  }
  return(list(Y=Y, Reject=R))
}

set.seed(12345)
Z <- accept_reject(n = 2000)
cat("The number of rejections: ",Z$Reject)
```

```
## The number of rejections:  629
```

```
hist(Z$Y, xlab = "Z", main = "Histogram of standard normal distribution", breaks = seq(-4,4,0.1))
```



The number of times required to the event "Acceptance" occured is a random variable which has a geometric distribution with success probability of:

$$p = Pr(U \leq \frac{f(Y)}{cg(Y)} | Y = y)$$

Considerind the density distribution $g(Y)$ and the value $\frac{f(y)}{cg(y)}$ we have:

$$p = \int_{-\infty}^{\infty} \frac{f(y)}{cg(y)} g(y) dy = \frac{1}{c} \int_{-\infty}^{\infty} f(y) dy = \frac{1}{c}$$

```
cat("Expected Rejection Rate: ER = ", 1 - (1/c))
```

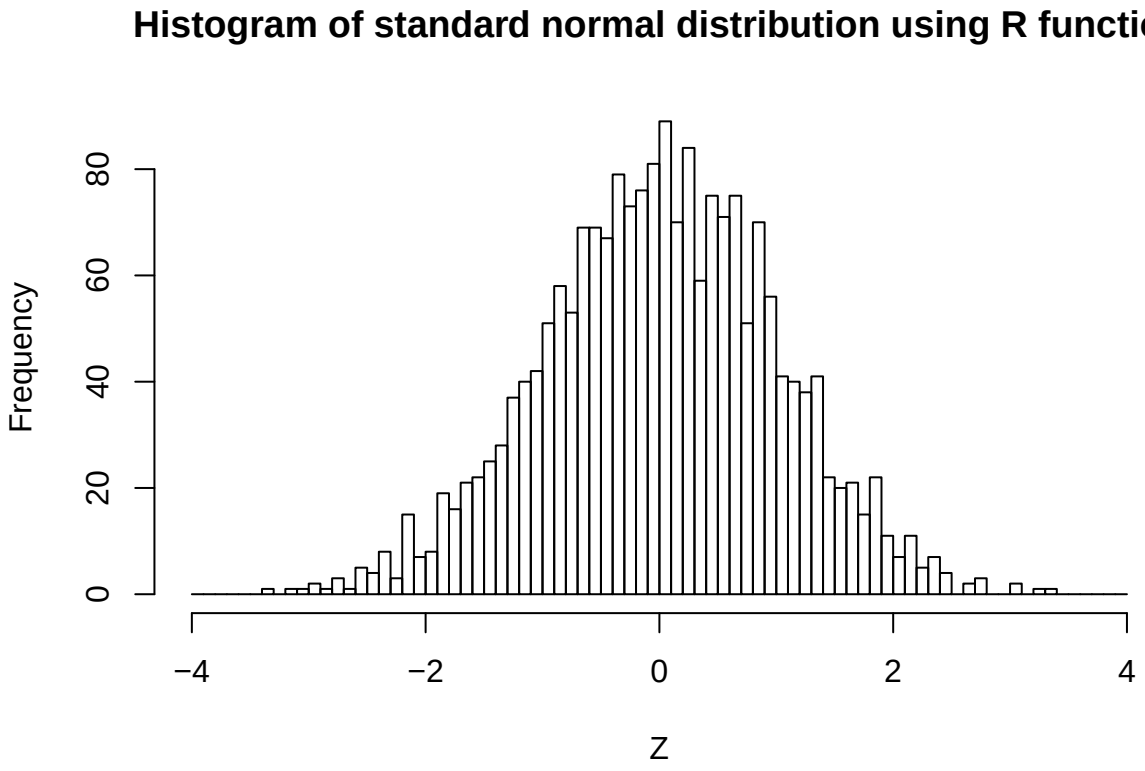
```
## Expected Rejection Rate: ER =  0.2398265
```

```
cat("Rejection Rate: R = ", Z$Reject/(Z$Reject+2000))
```

```
## Rejection Rate: R =  0.2392545
```

The rejection rate "R" is very close to the expected rejection rate "ER"

```
set.seed(12345)
Z = rnorm(2000)
hist(Z, main = "Histogram of standard normal distribution using R function", breaks = seq(-4,4,0.1))
```



- The histograms looks fairly similar i.e. generated from normal distribution

Appendix